Attorney Docket No. SOM920030008US1

# IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**Patent Application**

| | |
|---|---|
| Applicant(s): | A.J. Bradfield et al. |
| Docket No.: | SOM920030008US1 |
| Serial No.: | 10/699,036 |
| Filing Date: | October 31, 2003 |
| Group: | 2178 |
| Examiner: | Omar R. Abdul-Ali |

Title: Methods and Apparatus for Making Web
Browser Act Like Stand-Alone Application

## APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants (hereinafter referred to as "Appellants") hereby appeal the final rejection of claims 1, 3-11 and 14-20 of the above-identified application.

## REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, as evidenced by an assignment recorded October 31, 2003 in the U.S. Patent and Trademark Office at Reel 14666, Frame 969. The assignee, International Business Machines Corporation, is the real party in interest.

## RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences.

## STATUS OF CLAIMS

Claims 1, 3-11 and 14-20 stand finally rejected under 35 U.S.C. §103(a). Claims 2, 12 and 13 are canceled. Claims 1, 3-11 and 14-20 are appealed.

## STATUS OF AMENDMENTS

There has been no amendment filed subsequent to the final rejection.

## SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of: obtaining information from the information source; and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

An illustrative embodiment of the recited method of processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises the steps of: obtaining information from the information source (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101"); and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded (e.g., Specification, page 7, lines 12-13, "If Frame #1 is dependent on Frame #2, the code prevents the user from working on Frame #1 until Frame #2 is completely loaded"), said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded (e.g., Specification, page 8, lines 18-19, "If no dependency is found, the user can be allowed to interact with Frame #1 (step 211)").

Independent claim 4 recites a method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of: obtaining information from the information source; and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, wherein the preventing step further comprises rendering the displayed first portion inactive until after the second portion is sufficiently loaded.

An illustrative embodiment of the recited method of processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises the steps of: obtaining information from the information source (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101"); and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded (e.g., Specification, page 7, lines 12-13, "If Frame #1 is dependent on Frame #2, the code prevents the user from working on Frame #1 until Frame #2 is completely loaded"), wherein the preventing step further comprises rendering the displayed first portion inactive until after the second portion is sufficiently loaded (e.g., Specification, page 9, lines 21-22, "Until Frame #2 is fully loaded (step 303), Frame #1 is not active (i.e., dead frame), and Frame #2 continues to load (step 304)").

Independent claim 9 recites a method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of: obtaining information from the information source in accordance with an application, wherein the application comprises at least one subapplication; upon a request made to the subapplication, determining whether a current page of the application is loaded; when the current page is fully loaded, preserving data associated with the current page before loading the subapplication; and when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page.

An illustrative embodiment of the recited method of processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises the steps of: obtaining information from the information source in accordance with an application (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101), wherein the application comprises at least one subapplication (e.g., Specification, page 7, lines 18-19, "a web application has multiple subapplications"); upon a request made to the subapplication (e.g., Subapplication Requested 403 in FIG. 4), determining whether a current page of the application is loaded (e.g., Specification, page 11, lines 22-23, "the methodology checks whether the page has been fully loaded (step 403)"); when the current page is fully loaded, preserving data associated with the current page before loading the subapplication (e.g., Specification, page 11, lines 23-24, "If the page is fully loaded, the data is saved before loading the subapplication, and the requested subapplication is loaded (step 407)"); and when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page (e.g., Specification, page 11, line 26 through page 12, line 1, "if the page is not fully loaded, the requested subapplication is loaded directly without saving data, since the user has not entered any data (step 405)").

Independent claim 17 recites an apparatus for processing information, associated with an information source, in accordance with a browser, the apparatus comprising: a memory; and at least one processor coupled to the memory and operative to: (i) obtain information from the information source; and (ii) prevent a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

An illustrative embodiment of the recited apparatus (e.g., that described with reference to FIG. 5 in the Specification at page 12, lines10-26) for processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises: a memory (e.g., memory 502 in FIG. 5); and at least one processor (e.g., processor 501 in FIG. 5) coupled to the memory and operative to: (i) obtain information from the information source (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101); and (ii) prevent a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded (e.g., Specification, page 7, lines 12-13, "If Frame #1 is dependent on Frame #2, the code prevents the user from working on Frame #1 until Frame #2 is completely loaded"), said prevention occurring after a determination is made that the second portion depends on the first portion (e.g., Specification, page 8, line 18, "the methodology checks if Frame #1 depends on Frame #2 (step 207)"), otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded (e.g., Specification, page 8, lines 18-19, "If no dependency is found, the user can be allowed to interact with Frame #1 (step 211)").

Independent claim 18 recites an apparatus for processing information, associated with an information source, in accordance with a browser, the apparatus comprising: a memory; and at least one processor coupled to the memory and operative to: (i) obtain information from the information source in accordance with an application, wherein the application comprises at least one subapplication; (ii) upon a request made to the subapplication, determine whether a current page of the application is loaded; (iii) when the current page is fully loaded, preserve data associated with the current page before loading the subapplication; and (iv) when the current page is not fully loaded, load the subapplication without preserving data associated with the current page.

An illustrative embodiment of the recited apparatus (e.g., that described with reference to FIG. 5 in the Specification at page 12, lines10-26) for processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser(e.g., Web Browser 103 in FIG. 1), comprises: a memory (e.g., memory 502 in FIG. 5); and at least one processor (e.g., processor 501 in FIG. 5) coupled to the memory and operative to: (i) obtain information from the information source in accordance with an application (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101"), wherein the application comprises at least one subapplication (e.g., Specification, page 7, lines 18-19, "a web application has multiple subapplications"); (ii) upon a request made to the subapplication (e.g., Subapplication Requested 403 in FIG. 4), determine whether a current page of the application is loaded (e.g., Specification, page 11, lines 22-23, "the methodology checks whether the page has been fully loaded (step 403)"); (iii) when the current page is fully loaded, preserve data associated with the current page before loading the subapplication (e.g., Specification, page 11, lines 23-24, "If the page is fully loaded, the data is saved before loading the subapplication, and the requested subapplication is loaded (step 407)"); and (iv) when the current page is not fully loaded, load the subapplication without preserving data associated with the current page (e.g., Specification, page 11, line 26 through page 12, line 1, "if the page is not fully loaded, the requested subapplication is loaded directly without saving data, since the user has not entered any data (step 405)").

Independent claim 19 recites an article of manufacture for processing information, associated with an information source, in accordance with a browser, comprising a machine readable medium containing one or more programs which when executed implement the steps of: obtaining information from the information source; and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

An illustrative embodiment of the recited article of manufacture (e.g., Specification, page 13, lines 16-19) for processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises a machine readable medium (e.g., memory 502 in FIG. 5) containing one or more programs which when executed (e.g., by processor 501 in FIG. 5) implement the steps of: obtaining information from the information source (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101"); and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded (e.g., Specification, page 7, lines 12-13, "If Frame #1 is dependent on Frame #2, the code prevents the user from working on Frame #1 until Frame #2 is completely loaded"), said prevention occurring after a determination is made that the second portion depends on the first portion (e.g., Specification, page 8, line 18, "the methodology checks if Frame #1 depends on Frame #2 (step 207)"), otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded (e.g., Specification, page 8, lines 18-19, "If no dependency is found, the user can be allowed to interact with Frame #1 (step 211)").

Independent claim 20 recites an article of manufacture for processing information, associated with an information source, in accordance with a browser, comprising a machine readable medium containing one or more programs which when executed implement the steps of: obtaining information from the information source in accordance with an application, wherein the application comprises at least one subapplication; upon a request made to the subapplication, determining whether a current page of the application is loaded; when the current page is fully loaded, preserving data associated with the current page before loading the subapplication; and when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page.

An illustrative embodiment of the recited article of manufacture (e.g., Specification, page 13, lines 16-19) for processing information, associated with an information source (e.g., Web Page in Storage Unit 106 in FIG. 1), in accordance with a browser (e.g., Web Browser 103 in FIG. 1), comprises a machine readable medium (e.g., memory 502 in FIG. 5) containing one or more programs which when executed (e.g., by processor 501 in FIG. 5) implement the steps of: obtaining information from the information source in accordance with an application (e.g., Specification, page 6, line 19, "Code is sent from the server 104 to the client 101"), wherein the application comprises at least one subapplication (e.g., Specification, page 7, lines 18-19, "a web application has multiple subapplications"); upon a request made to the subapplication (e.g., Subapplication Requested 403 in FIG. 4), determining whether a current page of the application is loaded (e.g., Specification, page 11, lines 22-23, "the methodology checks whether the page has been fully loaded (step 403)"); when the current page is fully loaded, preserving data associated with the current page before loading the subapplication (e.g., Specification, page 11, lines 23-24, "If the page is fully loaded, the data is saved before loading the subapplication, and the requested subapplication is loaded (step 407)"); and when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page (e.g., Specification, page 11, line 26 through page 12, line 1, "if the page is not fully loaded, the requested subapplication is loaded directly without saving data, since the user has not entered any data (step 405)").

## GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

(I) Whether claims 1, 3-9, 11 and 14-20 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent Application Publication No. 2003/0018717 (hereinafter "Mikhailov").

(II) Whether claim 10 is unpatentable under 35 U.S.C. §103(a) over Mikhailov in view of U.S. Patent Application Publication No. 2002/0152110 (hereinafter "Stewart").

## ARGUMENT

Appellants incorporate by reference herein the disclosure of their previous response filed in the present application, namely the response dated August 30, 2007.

(I) Whether claims 1, 3-9, 11 and 14-20 are unpatentable under 35 U.S.C. §103(a) over U.S. Patent Application Publication No. 2003/0018717 (hereinafter "Mikhailov").

With regard to the §103(a) rejections, Appellants initially note that a proper case of obviousness requires that the cited references when combined must "teach or suggest all the claim limitations," and that there be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine the references or to modify the reference teachings. See Manual of Patent Examining Procedure (MPEP), Eighth Edition, August 2001, §706.02(j).

Appellants submit that the Examiner has failed to establish a proper case of obviousness in the §103(a) rejection of claims 1, 3-9, 11 and 14-20 over Mikhailov, in that the Mikhailov reference, even if assumed to be combinable, fails to teach or suggest all the claim limitations, and in that no cogent motivation has been identified for modifying the reference teachings to reach the claimed invention.

Claim 1 is directed to a method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of: obtaining information from the information source; and preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

In an illustrative embodiment of the present invention, FIG. 2, a flow diagram illustrates a client-based methodology 200 for avoiding errors when interacting with partially loaded pages, according to a first embodiment of the present invention. A live frame is shown to the user at the client (via monitor 102), however, interactions are intercepted and the user is asked to try again (to interact) when the page fully loads. A "live frame" refers to a frame that is active, i.e., the user can interact with (e.g., click on hyperlinks, enter data, select items, etc.).

When the user sends a request (step 201) to the server 104 to access a web page, the page begins loading (step 202). In some special case, Frame #1 depends on Frame #2's information to continue (step 207). It is very common that Frame #1 is already activated, and Frame #2 is still in the process of loading (step 208). To determine if the user can continue working on Frame #1, the methodology checks if Frame #1 depends on Frame #2 (step 207). If no dependency is found, the user can be allowed to interact with Frame #1 (step 211). If a dependency exists between Frame #1 and Frame #2, a check is made to determine whether Frame #2 is fully loaded (step 208). If loaded, the user is allowed to work with Frame #1 (step 211) and so on. If not loaded, the user is instructed via an alert message to wait (step 209), while Frame #2 continues loading (step 210). The browser 103 continues to load more data from Frame #2. If the user tries to interact with Frame #1 (step 204) during this process, e.g., a subapplication is requested (step 205), an alert message pops up to tell the user to wait until the page is fully loaded (step 206). Once the user has addressed the alert message, Frame #2 keeps loading (step 203). This process continues until the page has completely loaded (end block 212). Thus, advantageously, if Frame #1 and Frame #2 are fully loaded or if Frame #1 does not depend on Frame #2, then the user is allowed to interact with Frame #1. However, if Frame #2 depends on Frame #1 and Frame #2 is not fully loaded, or if the user requests a subapplication (e.g., associated with Frame #2), then the user is instructed to wait until Frame #2 fully loads.

The Examiner in formulating the §103(a) rejection of claim 1 argues that each and every one of the above-noted limitations of claim 1 is met by the teachings of Mikhailov. Below, Appellants explain how such portions of Mikhailov fail to teach or suggest what the Examiner contends that Mikhailov teaches or suggests.

In characterizing the Mikhailov reference as allegedly meeting certain limitations of claim 1, the Examiner relies primarily on paragraphs [0008] and [0017] of Mikhailov. However, the relied-upon portions of Mikhailov fail to teach or suggest the limitations as alleged.

Mikhailov at paragraph [0008] refers to blocking immediately an application after the user initiates a page submission to the server and the application remains unusable until the page that the server sent in response is loaded, processed and rendered on the screen. Such blocking mode of operation leaves users unproductively waiting every time a submission or request is made to the server. Mikhailov teaches a method of handling mobile data to enable application users to be more productive. Mikhailov implements stateful frame navigation. The system receives a first command selecting a first icon and, in response, links to a first site and displays a first frame in an active mode for user interaction with the first site. When the system receives a command selecting a second icon and, in response, links to a second site, the system deactivates the first frame from the active mode, and displays a second frame in the active mode for user interaction with the second site. To enable application users to be more productive, the system taught in Mikhailov maintains the first frame in a background mode which preserves the altered state of the first frame, while the second frame is in the active mode. In response to receiving a second command selecting the first icon the system links to the first site and displays the first frame in the active mode.

However, Mikhailov does not determine if a second frame depends on the first frame before having the first frame deactivated and maintained in a background mode. Furthermore, Mikhailov does not teach or suggest permitting the user to interact with the displayed first frame regardless of whether the second frame is sufficiently loaded when it is determined that the second frame does not depend on the displayed first frame. Thus, Mikhailov does not teach or suggest "preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded."

In response to Appellants' arguments, the Examiner refers to paragraph [200] of Mikhailov as teaching or suggesting permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded after a determination is made that the second portion does not depend on the first portion. Appellants respectfully disagree. The relied upon portion of Mikhailov discloses that the user can still navigate between frames, activate other frames and continue browsing and interact with the content available in other frames. The user cannot navigate to the frame being loaded, since the blocked status and browsing restrictions are applied to the content area 3001 of the frame being loaded.

Accordingly, it is believed that the teachings of Mikhailov fail to meet the limitations of claim 1.

Also, the Examiner provides the following statement of motivation beginning at page 2, last paragraph of the Office Action:

> Mikhailov does not explicitly disclose permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded, however it would have been obvious to one having ordinary skill in the art at the time the invention was made to do so. One would have been motivated to permit the user to interact with the displayed first portion if there is no dependency between the two frames so that the user can continue working on the displayed frame without interruption.

Appellants respectfully submit that this is a conclusory statement of the sort rejected by both the Federal Circuit and the U.S. Supreme Court. See KSR v. Teleflex, No. 13-1450, slip. op. at 14 (U.S., Apr. 30, 2007), quoting In re Kahn, 441 F. 3d 977, 988 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness."). There has been no showing in the present §103(a) rejection of claim 1 of objective evidence of record that would motivate one skilled in the art to produce the particular limitations in question. The above-quoted statement of motivation provided by the Examiner appears to be a conclusory statement of the type ruled insufficient in KSR v. Teleflex.

Independent claims 17 and 19 include limitations similar to those of claim 1, and are therefore believed allowable for reasons similar to those described above with reference to claim 1.

Dependent claims 3, 5-8 are allowable for at least the reasons identified above with regard to claim 1. One or more of these claims are also believed to define separately-patentable subject matter over the cited art.

Claim 9 is directed to a method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of: obtaining information from the information source in accordance with an application, wherein the application comprises at least one subapplication; upon a request made to the subapplication, determining whether a current page of the application is loaded; when the current page is fully loaded, preserving data associated with the current page before loading the subapplication; and when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page.

In an illustrative embodiment of the present invention, FIG. 4, a flow diagram illustrates a client-based methodology 400 for preserving state. Suppose a very complicated web application has several subapplications. The user can jump back and forth among these subapplications. In one of these subapplications, assume that there is a page that has a large amount of data entered by the user. The present invention is used to save the data in the page before jumping to other subapplications. Later, when the user returns, the entire data the user typed in can be recovered.

As shown in FIG. 4, a user interacts (e.g., accesses different page/subapplication) with a page containing a large amount of data which needs to be preserved (step 401). The page starts to load (step 402). At the point of user interaction (e.g., subapplication request in step 403), the methodology checks whether the page has been fully loaded (step 404). If the page is fully loaded, the data is saved before loading the subapplication, and the requested subapplication is loaded (step 407). When the user comes back from a subapplication, all of the data that was saved is recovered (step 408). Going back to step 404, if the page is not fully loaded, the requested subapplication is loaded directly without saving data, since the user has not entered any data (step 405). Thus, the subapplication is loaded regardless of whether the user entered data. If the data was not saved before the subapplication was loaded, the page is displayed directly (step 406).

Mikhailov fails to teach or suggest the limitations of claim 9. Although Mikhailov, in paragraph [0018] teaches of preserving the altered state of a first frame in the background while a second frame is active, Mikhailov does not differentiate between a fully loaded and a not fully loaded first frame before preserving the state of the first frame. Also, Mikhailov does not teach or suggest not preserving data associated with the first frame if the first frame had not been fully loaded before the system received a command selecting the activation of a second frame.

Accordingly, it is believed that the teachings Mikhailov fails to meet the limitations of claim 9.

Also, Examiner provides the following statement of motivation beginning at page 5, last paragraph of the Office Action:

> However, it would have been obvious to one having ordinary skill in the art at the time the invention was made to refrain from preserving data of a current page when linking to another site. One would have been motivated to load the subapplication without preserving data associated with the current page to improve computer efficiency. Loading both pages simultaneously would reduce processing speed, so it would be advantageous to load the second page without preserving the data from the first frame.

The above-quoted statement of motivation provided by the Examiner appears to be a conclusory statement of the type ruled insufficient in KSR v. Teleflex.

For at least these reasons, Appellants assert that claim 9 is patentable over Mikhailov.

Independent claims 18 and 20 include limitations similar to those of claim 9, and are therefore believed allowable for reasons similar to those described above with reference to claim 9.

Dependent claims 11 and 14-16 are allowable for at least the reasons identified above with regard to claim 9. One or more of these claims are also believed to define separately-patentable subject matter over the cited art.

Regarding claim 3, Mikhailov does not disclose instructing the user to wait to interact with the first portion until after the second portion is sufficiently loaded, when the determination is made that the second portion depends on the first portion. Appellants disagree with the Examiner's statement that "it would have been obvious to one having ordinary skill in the art at the time the invention was made to include a message instructing the user to wait to interact with the first portion until the second portion of the page is sufficiently loaded. One would have been motivated to include a message instructing the user to wait in order to notify the user that the second page is loading and interaction is disabled." (See page 3, second paragraph of the final Office Action).

(II) Whether claim 10 is unpatentable under 35 U.S.C. §103(a) over Mikhailov in view of U.S. Patent Application Publication No. 2002/0152110 (hereinafter "Stewart").

With regard to the §103(a) rejection of claim 10, claim 10 is allowable for at least the reasons identified above with regard to claim 9. Also, claim 10 is believed to define separately-patentable subject matter over the cited art. Stewart fails to remedy the deficiencies described above with regard to Mikhailov. Accordingly, it is believed that the combined teachings of Stewart and Mikhailov fail to meet the limitations of claim 10.

In view of the above, Appellants believe that claims 1, 3-11 and 14-20 are in condition for allowance, and respectfully request withdrawal of the §103(a) rejection.

Respectfully submitted,

William E Lewis

Date: November 30, 2007

William E. Lewis
Attorney for Appellant(s)
Reg. No. 39,274
Ryan, Mason & Lewis, LLP
90 Forest Avenue
Locust Valley, NY 11560
(516) 759-2946

APPENDIX

1. A method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of:

obtaining information from the information source; and

preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

3. The method of claim 1, wherein the preventing step further comprises instructing the user to wait to interact with the first portion until after the second portion is sufficiently loaded, when the determination is made that the second portion depends on the first portion.

4. A method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of:

obtaining information from the information source; and

preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, wherein the preventing step further comprises rendering the displayed first portion inactive until after the second portion is sufficiently loaded.

5. The method of claim 1, wherein the second portion is sufficiently loaded when it is fully loaded.

6. The method of claim 1, wherein the browser is implemented on a client computer system.

7. The method of claim 1, wherein the browser comprises a web browser.

8. The method of claim 1, wherein the information source comprises at least one server computer system.

9. A method of processing information, associated with an information source, in accordance with a browser, the method comprising the steps of:

obtaining information from the information source in accordance with an application, wherein the application comprises at least one subapplication;

upon a request made to the subapplication, determining whether a current page of the application is loaded;

when the current page is fully loaded, preserving data associated with the current page before loading the subapplication; and

when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page.

10. The method of claim 9, wherein the preserving step further comprises storing the user provided data in at least one of a session object and a hidden frame.

11. The method of claim 9, further comprising the step of retrieving the preserved data when the user returns from the subapplication.

14. The method of claim 9, wherein the browser is implemented on a client computer system.

15. The method of claim 9, wherein the browser comprises a web browser.

16. The method of claim 9, wherein the information source comprises at least one server computer system.

17. Apparatus for processing information, associated with an information source, in accordance with a browser, the apparatus comprising:

a memory; and

at least one processor coupled to the memory and operative to: (i) obtain information from the information source; and (ii) prevent a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

18. Apparatus for processing information, associated with an information source, in accordance with a browser, the apparatus comprising:

a memory; and

at least one processor coupled to the memory and operative to: (i) obtain information from the information source in accordance with an application, wherein the application comprises at least one subapplication; (ii) upon a request made to the subapplication, determine whether a current page of the application is loaded; (iii) when the current page is fully loaded, preserve data associated with the current page before loading the subapplication; and (iv) when the current page is not fully loaded, load the subapplication without preserving data associated with the current page.

19. An article of manufacture for processing information, associated with an information source, in accordance with a browser, comprising a machine readable medium containing one or more programs which when executed implement the steps of:

obtaining information from the information source; and

preventing a user from interacting with a displayed first portion of the received information until after a second portion of the received information is sufficiently loaded, said prevention occurring after a determination is made that the second portion depends on the first portion, otherwise, permitting the user to interact with the displayed first portion regardless of whether the second portion is sufficiently loaded.

20. An article of manufacture for processing information, associated with an information source, in accordance with a browser, comprising a machine readable medium containing one or more programs which when executed implement the steps of:

obtaining information from the information source in accordance with an application, wherein the application comprises at least one subapplication;

upon a request made to the subapplication, determining whether a current page of the application is loaded;

when the current page is fully loaded, preserving data associated with the current page before loading the subapplication; and

when the current page is not fully loaded, loading the subapplication without preserving data associated with the current page.

## EVIDENCE APPENDIX

None.

## RELATED PROCEEDINGS APPENDIX

None.